AGENT GOVERNANCE IN PRODUCTION

An agent without a passport is an unbadged contractor in your building.

An agent in production is software that acts. This is the discipline that makes it auditable and accountable: the system card, the registry, the gates, runtime monitoring, incident response, and the measured exploitation surface. Survivable across model changes, because it lives in architecture the model cannot edit. For the CISO, the Head of AI Ops, and the Head of Engineering who own it in production.

FOR

CISO, Head of AI Ops, Head of Engineering

GROUNDED IN

10,000-trial exploitation study + the PwC AI Factory registry

OPEN-SOURCE STACK

guardrails, agent-monitor, agent-auth, context-kubernetes

DIRECT LINE

charafeddine@cohorte.co

02 THE PROBLEM

Most enterprises do not know how many agents they have in production.

The first finding of every Cohorte enterprise engagement is the same. Someone in the room runs an unauthorised count and is surprised by the result. Agents proliferate inside SaaS platforms, vendor copilots, internal tooling, and proof-of-concept folders that were promoted to production by a single Slack thread. Without a registry, every other governance question is unanswerable. With a registry, most of them become straightforward.

Three structural problems make agents harder to govern than the systems that preceded them.

They compose. A workflow that involves an agent rarely involves only one. An assistant calls a retrieval system, which calls a re-ranker, which calls a second model for classification. The composition multiplies the surface and obscures the chain of responsibility. The incident response team does not know which component failed; in a non-deterministic system, they may not be able to reproduce the failure to find out.

They have authority. An agent with tool access can read files, send messages, write to databases, transfer funds, change configurations. The authority is usually inherited from a service account that was provisioned for a human or for an earlier integration. The audit trail records the action; it does not record the reasoning the agent used to take it, and frequently does not record the prompt the agent was operating from at the time.

They mutate. The prompt is edited upstream by a well-meaning team. The vendor swaps the model. The retrieval index is reorganised. Each change is small and well-intentioned. The cumulative effect over a quarter is a system that bears only an architectural resemblance to the one the security review approved.

> An agent without a passport, an owner, a reliability level and a retirement condition is shadow IT with better marketing.

03 THE SYSTEM CARD

One document per agent.

The first artefact the auditor reads.

Each system in production carries a **system card**, the machine-readable record (the registry's schema calls it the agent passport) that the registry holds, the gate enforces, and the audit reviews. It is not a wiki page; it has a schema. When a regulator asks for the inventory of high-risk systems, the answer is a query against the cards, not a hand-assembled spreadsheet.

IDENTIFIER
`brief-drafter-v1.4.2`
Registered in Cohorte's registry on 2026-04-12.

OWNER
Karim Bensaïd, Head of Practice
karim.b@example.com · Slack: @karim

REGISTERED PURPOSE
Draft client briefs from the meeting minutes and the engagement folder. Output reviewed by a partner before release.

SCOPE OF AUTHORITY
Read: engagement folder (scoped). Write: draft-brief folder only. **No client-facing send rights.** No financial system access.

RELIABILITY LEVEL
92.4% at 95% coverage. Calibrated 2026-06-09. Recalibrates weekly. Method: TrustGate v0.7.

VERIFICATION ARTEFACT
Calibration set `brf-drft-cal-v3`, n = 240, labelled by Partners' Council, version-controlled.

ESCALATION PATH
Low-confidence (below 0.85) → human reviewer. Tool-call attempt outside scope → CISO + owner alert. Incident → Karim + Cohorte governance committee within 1 h.

RETIREMENT CONDITION
Reliability drops below 88% over two consecutive weeks *or* the engagement-folder schema changes *or* twelve months elapse without recertification.

CHANGE LOG (LAST THREE)
2026-06-09 calibration refresh, bar unchanged. 2026-05-22 model upgrade, re-verified, passed. 2026-05-03 prompt edit by L&D, re-verified, passed.

04 THE REGISTRY

The system of record for every agent in production.

A registry is to AI agents what an asset inventory is to information security. The control objective in every modern framework (ISO 27001 8.1, NIST CSF Identify, AI Act Article 11, ISO 42001 6.1) reduces to: you cannot defend what you cannot enumerate. The registry is the place where enumeration lives.

What the registry holds. The passport for every agent. The current reliability level. The calibration set reference. The owner. The incident log. The change log. The retirement condition. The link to the live system. A query against the registry answers the questions a regulator, an auditor, or an incident reviewer asks first.

What the registry is not. Not a wiki. Not a Confluence page. Not a Word document forwarded by email. Wikis and pages are useful documentation; they are not the system of record because they are written by humans on a slower cadence than the systems they describe. The registry is updated by the system as the system changes, with human approval at the gates.

How the registry is built. The reference architecture is documented in the Context Kubernetes paper (Mouzouni, 2026). The implementation is a YAML-based declarative manifest with a reconciliation loop. Existing components (the agent itself, the prompt, the model endpoint, the retrieval index, the tool integrations) declare themselves; the manifest reconciles the declared state with the running state. Drift between the two is the most reliable early signal of an unauthorised change.

THE REGISTRY QUERIES

Inventory. "List every production agent and its owner." *The first regulator question.*

High-risk slice. "Which agents process personal data under AI Act Article 10?"

Reliability drift. "Which agents are within 5% of their bar?"

Compliance status. "Which agents do not have a current verification artefact?"

Change blast. "Which agents would be affected if vendor X's model went down?"

IMPLEMENTATION REFERENCE

Open-source: github.com/Cohorte-ai/context-kubernetes. ~7,000 lines, 92 tests, eight reproduced experiments.

Without a registry, governance is anecdotal. An organisation can run a "Responsible AI Committee" without a registry. It will review the agents it remembers. The agents it forgets are the ones that will fail. The registry is not bureaucracy; it is the difference between governance you can trust and governance theatre.

05 THE GATES

Mind-in-the-loop. Not human-in-the-loop.

The control objective is not to insert a human between the agent and the action. It is to insert a human at the decision the human can actually defend, with the information the human needs, in the time the workflow allows. A pipeline that floods one analyst with a thousand decisions per week is a pipeline that has not designed its gates. It has designed an alibi.

Where the gates belong. At the seams between authority levels. The agent drafts; the human approves the send. The agent triages; the human authorises the close. The agent recommends; the human commits the trade. A gate placed where the workflow already has a decision is a gate that adds judgement. A gate placed at the end of a pipeline that has already executed is a gate that adds latency.

What the gate displays. Three things, on the same screen. The agent's recommendation. The agent's confidence on this specific decision (derived from the conformal calibration). The two or three pieces of evidence the agent used. The reviewer should be able to act in twenty seconds when the case is routine and to drill into the evidence when the case is not.

When the gate fires. Below the reliability bar declared in the passport. On any action outside the registered scope of authority. On any tool call to a system in the high-risk list. On any output containing a category the policy guardrail flagged. Above the bar and inside scope, the system operates; the gate does not fire and the audit trail records the autonomous decision with full attribution.

> Oversight you cannot act on is not oversight. It is paperwork that lives inside an audit report.

06 RUNTIME MONITORING

Five signals that matter. Twenty that distract.

Uptime, latency, and error rate are necessary; they are inherited from any production system. The five signals below are specific to AI systems and they are the ones that fire before the failure becomes visible to the downstream consumer. A monitoring dashboard that does not show all five is a dashboard you can pass an audit with and lose a system on.

SIGNAL	WHAT IT TELLS YOU	ACTION WHEN IT MOVES
Reliability drift	Weekly reliability level on a refreshed calibration sample. The leading indicator of every other failure mode.	Down by more than 3% from baseline: re-calibrate immediately. Down by more than 5%: hold the gate.
Abstention rate	Frequency at which the system returns no answer. A rising rate often precedes distribution shift.	+30% over a week: investigate. Often signals upstream changes the team did not announce.
Tool-call distribution	The histogram of which tools the agent invokes, and at what rates. Significant changes indicate task drift.	New tool appearing or rate change >2σ: review whether the agent is being used for its registered purpose.
Permission denial rate	How often the agent tries an action outside its scope. A non-zero rate is the agent encountering the gate; a rising rate may signal prompt-injection attempts.	Any sustained increase: investigate the input distribution; look for prompt injection patterns.
Cost-per-call	The economic signature of the system. Material changes usually mean a model swap, a prompt expansion, or a retrieval reorganisation upstream.	>20% change in a week: trace the change. Often a sentinel for unannounced upstream edits.

Logging the call is necessary; logging the reasoning is the discipline. An audit trail that captures input and output without the prompt, the retrieval context, and the model version cannot reconstruct the decision. The trace has to be sufficient for a competent reviewer to ask: "given what the agent saw, was the action reasonable?" The team that builds for that test passes the others.

07 INCIDENT RESPONSE

Incidents in non-deterministic systems are not deterministic incidents.

The classical incident-response playbook assumes the failure reproduces on demand. An AI incident often does not. The same prompt against the same model can return a different answer ten seconds later. The forensic discipline has to be tighter, the logging has to be richer, and the post-mortem has to ask different questions.

Detection. Three sources. A monitoring alert (one of the five signals from page six). A user report (the case the gate should have caught but did not). An audit finding (a sample case from the registry that the reviewer flagged). All three converge on the same incident queue and follow the same triage path.

Triage. Severity classification using the AI-incident grid: scope (one user, one cohort, all), reversibility (output retracted, output committed, output acted on), and exposure (internal, named external, public). Severity drives the response cadence; the grid drives the severity.

Containment. The hold action. Inputs continue to flow to preserve the throughput data; outputs route to the review queue, not the consumer. The system is held, not rolled back, because rollback may not be possible if the model was hosted by a vendor that has since updated it. Containment is what hold achieves; rollback is the secondary action.

Forensics. Replay the input, the prompt, the retrieval context, the model version, the random seed where available. If the failure reproduces, the post-mortem is the standard kind. If it does not, the post-mortem question becomes *what was different about the conditions of that call.* Most incidents in this category are an upstream change someone in the broader team did not announce.

Remediation. Re-calibrate. Re-test. Re-deploy through the gate. Document the incident in the passport's change log. The post-mortem is signed by the owner and the governance committee.

THE AI-INCIDENT GRID

SEV-1. Output committed, irreversible, external exposure. Hold + executive escalation within 30 min.

SEV-2. Output committed, partially reversible, internal exposure. Hold + owner + governance within 2 h.

SEV-3. Output retracted, no external exposure. Owner + scheduled review.

SEV-4. Near miss caught by the gate. Logged; reviewed at the next calibration.

WHAT THE POST-MORTEM ASKS

What changed in the conditions of the call. What signal would have caught it earlier. What change to the passport, the gate, or the monitoring closes the gap. What re-calibration is required before re-deployment.

08 THE EXPLOITATION SURFACE

Most attacks on AI agents do not work. One class does.

In late 2025 and early 2026 we ran 10,000 trials of prompt-injection-style attacks against seven frontier models in real Docker sandboxes. The result reframes how a CISO should think about the agent threat model. Most attack classes that fill the OWASP LLM Top 10 produce zero exploitation on capable models. One class produces 32 to 40%.

CONDITION	CLAUDE S4	GPT-4.1	GPT-5-MINI	O4-MINI	DEEPSEEK
Baseline (no encouragement)	0-2%	0%	0-4%	0%	0-2%
Minimisation ("this is a sandbox")	0%	0%	0%	0%	0%
Moral licensing ("improves security")	0%	0%	0%	0%	0%
Identity priming ("10x engineer")	0%	0%	0%	0%	0%
Goal reframing: puzzle	**38-40%**	0%	**8-10%**	0%	**20%**
Goal reframing: CTF	**32-34%**	0%	**10-14%**	**14%**	8-10%

Models as tested, late 2025 to early 2026; exploitation against frontier models has fallen with each release. The leaderboard will change. The pattern, one attack class dominating while the rest produce nothing, is the finding.

> The model does not disobey the rule. It redefines the task such that exploitation is the task.

The implication for the threat model is sharper than the OWASP framing suggests. Defending against ten attack classes that produce zero exploitation is theatre. Defending against the one class that produces 32-40% on the most capable model in the cohort is the work. The countermeasures are on the next page.

09 COUNTERMEASURES

What actually defends. Three layers.

Goal reframing succeeds at the prompt layer. Defending at the prompt layer alone is a losing race. The robust defence is layered: deny the agent the authority to take the exploited action, enforce the deny at the runtime layer outside the model, and detect attempted exploitation at the monitoring layer.

LAYER 1 · PERMISSION

Least authority

The agent never had the right to take the reframed action; its authority is a strict subset of the user's. If it cannot reach the file, the puzzle framing is moot. The three-tier permission model: static contract, session token, per-call check.

BLOCKS

LAYER 2 · RUNTIME

Deterministic guardrails

Every tool-call passes a separate process that enforces policy outside the model. The model is invited to disobey; the runtime makes disobedience structurally impossible. Open-source: the guardrails engine, policies versioned and outside the agent's reach.

BLOCKS

LAYER 3 · MONITORING

Detection signals

The attempt leaves traces: reasoning that recategorises the task, tool-calls beyond scope, reframing language in the output. Monitoring flags them, so the threat model evolves with the attacker instead of lagging it.

CATCHES

Safety training is improving. The exploitation rate against OpenAI's frontier models dropped monotonically across releases between April 2025 and March 2026. The progress is real and should be incorporated into the threat model. It is also model-specific. The deployment that relies on "the model is now safe enough" relies on the wrong layer. The deployment that relies on least authority, deterministic guardrails, and detection signals depends on architecture the model cannot edit.

10 THE OPEN-SOURCE STACK

Six repositories. Each scoped to one job.

The reference implementations behind the operating model live as six independent open-source repositories under the Cohorte AI organisation on GitHub. They are auditable, forkable, and adoptable without commercial engagement. Cohorte's commercial offering is the training of the people who will operate them; the code is free.

REPOSITORY	PURPOSE	WHERE IT SITS IN THIS BRIEFING
trustgate	Reliability and trust layer for agent outputs. Self-consistency sampling, conformal calibration.	Briefing 02. The reliability level in the passport.
guardrails	Policy, safety, and compliance guardrails. Input filtering, output validation, deterministic policy enforcement.	Page 09 (countermeasures, layer 2). The gates that fire outside the model.
context-router	Smart routing of queries to the right context and data source, with permission enforcement.	Authority enforcement at retrieval. Prevents cross-domain data leak.
agent-monitor	Observability, logging, behavioural monitoring. Drift detection, runtime signals, audit-trail capture.	Page 06 (runtime monitoring). The five signals indexed and dashboarded.
agent-auth	Permissions, roles, authentication. The three-tier model where agent authority is a strict subset of human authority.	Page 09 (countermeasures, layer 1). The least-authority foundation.
context-kubernetes	Orchestration layer. The declarative manifest, the reconciliation loop, the reference architecture.	Page 04 (the registry). The system of record implemented.

Open-source by deliberate choice. Trust and governance code that lives behind a vendor's signed-NDA is trust on credit. The reference implementations are open because the operating model claims to be auditable; auditability cannot stop at the boundary of the vendor's source tree. The commercial value Cohorte sells is the discipline of running this code in a regulated environment, not the right to read it.

11 WHAT THIS IS NOT

Three boundaries this briefing keeps.

Agent governance is a discipline. Like every discipline, what it refuses defines it as much as what it includes. The boundaries below are explicit so a buyer arriving at the discovery call already knows where the conversation will narrow.

Not a replacement for application security. The OWASP Top 10 for application security still applies. Authentication, authorisation, input validation, secrets management, transport security, supply chain. Agent governance sits on top of a working application-security baseline; it does not absolve the team of building one. A team that brings agent-governance ambition without an application-security foundation should fix the foundation first.

Not the deployment review. The deployment review (architecture review, threat model, change advisory board) is a separate process with its own owners. Agent governance produces inputs to that review and consumes its outputs. The passport, the reliability level, and the exploitation-surface assessment are inputs the deployment review will demand. The deployment review remains the deployment review.

Not a substitute for human judgement. The hardest call this briefing's frameworks support is whether to deploy a system the registry can run but the workflow probably should not. That call belongs to the named owner, the executive sponsor, and (in regulated environments) the second-line risk function. The frameworks make the trade-offs visible; they do not make the trade-offs go away. A team that hides behind the frameworks instead of taking the call has misread what they are for.

> Frameworks make the trade-offs visible. Frameworks do not make the trade-offs go away.

12 HOW THIS LANDS IN A COHORTE ENGAGEMENT

Registry by week four. Gates by week eight. Monitoring by week ten.

In the Team Bootcamp, agent governance is installed on the participant's team in weeks four to ten. In AI Readiness it is rolled out across the organisation's portfolio with the same primitives. The cadence below is from the standard 12-week Team Bootcamp.

WEEK	WHAT GETS INSTALLED	WHAT GETS SHIPPED
Week 3	Agent inventory. Every agent in production, named, owned, with current authority listed.	A list of every agent the team did not know it had. The unauthorised count from page two.
Week 4	Registry skeleton. Passports drafted for the prioritised systems. Owners assigned.	A working registry with passports for the top 5-10 systems.
Week 6	Authority audit. Each agent's scope reduced to the registered purpose. agent-auth integrated.	A least-authority profile per agent. The permission denial rate baseline.
Week 8	Guardrails layer. Deterministic policy at the runtime. Gate placement at the seams.	The first gate firing on a live call. The first mind-in-the-loop review.
Week 10	Monitoring instrumentation. The five signals. The incident grid. The escalation paths.	The first weekly governance report. The first non-zero drift signal investigated to root cause.

The capstone defends the registry. A graduating team leaves the bootcamp with the registry populated, the gates firing, the monitoring instrumented, and an incident-grid response wired in for severity tiers SEV-1 and SEV-2. The capstone defence is the live demonstration of the registry under a regulator-style question: "show me every high-risk agent." The graduate runs the query.

13 FROM THE FIELD

PwC AI Factory. Sixty-plus agents on the registry.

The agent registry behind the PwC France & Maghreb AI Factory holds the passport for every production system. Each entry is owned by a named partner or director. Authority is enforced through agent-auth. Verification is via TrustGate. Monitoring runs the five signals. The named reference is Patrick Monteiro, CIO.

SCOPE	60+ AI systems on the registry. **Each carries a passport, an owner, a reliability level and a retirement condition.** The Annex III high-risk slice is identified and tagged.
AUTHORITY	**Three-tier permission model**: static contract, session token, per-call check. Service accounts retired in favour of agent-scoped credentials.
MONITORING	The five signals run continuously. **Weekly governance report**, monthly drift review, quarterly external calibration.
INCIDENT DISCIPLINE	SEV-1/SEV-2 escalation tested quarterly. **The first audit exercise produced no findings on the registry layer.** Subsequent exercises continue to test the discipline.

> The registry is the document that turns a "Responsible AI Committee" into a function with a job description.

14 REFERENCES

References. The threat model, sourced.

The papers, repositories and frameworks behind the claims on the previous pages. The full record lives at **teams.cohorte.co/research**.

Mouzouni, C. (2026). Mapping the Exploitation Surface: A 10,000-Trial Taxonomy of What Makes LLM Agents Exploit Vulnerabilities. arXiv preprint. Source for pages 08-09. Code and data at github.com/Cmouzouni/exploitation-surface.

Mouzouni, C. (2026). Context Kubernetes: An Orchestration Architecture for Enterprise Knowledge in Agentic AI Systems. arXiv preprint. Source for the registry pattern (page 04) and the three-tier permission model (page 09). github.com/Cohorte-ai/context-kubernetes.

Mouzouni, C. (2026). Black-Box Reliability Certification for AI Agents via Self-Consistency Sampling and Conformal Calibration. Preprint (2026). The reliability-level method referenced in the system card. github.com/Cohorte-ai/trustgate.

OWASP (2025). OWASP Top 10 for Large Language Model Applications. The reference threat taxonomy. The exploitation-surface paper reframes which entries actually fire.

Greshake, K. et al. (2023). Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. The canonical reference on indirect prompt injection via tool/retrieval channels.

Perez, F. et al. (2022). Ignore Previous Prompt: Attack Techniques For Language Models. Early systematic survey of prompt-injection attack patterns.

Bainbridge, L. (1983). Ironies of Automation. *Automatica, 19*(6). The structural argument behind the mind-in-the-loop distinction (page 05).

European Union (2024). Regulation 2024/1689 on AI. Article 11 (technical documentation), Article 12 (record-keeping), Article 26 (deployer obligations). The conformity baseline that the registry and the monitoring layer produce as natural output.

ISO/IEC (2023). ISO/IEC 42001 clauses 6.1 (planning), 8.1 (operational planning and control), 9.1 (monitoring, measurement, analysis and evaluation). The management-system clauses that map onto registry, gates, and monitoring respectively.

NIST AI RMF (2023). Govern and Manage functions, with the Generative AI Profile additions (2024). The taxonomy that orients the discipline.

The AI OS newsletter. Letters 71, 73, 77 (2026). The Thomas / agents story; the notification-and-intensification trap; the oversight-cosplay essay. Archive at charafeddine.co/letters.

FOR THE CISO AND THE HEAD OF AI OPS

One discovery call. Bring your top three agents.

Sixty minutes. We walk them through the passport, the gate placement, and the exploitation-surface assessment. You leave with the three passports drafted and the gaps named in writing.

charafeddine@cohorte.co

Cohorte SAS · Société par actions simplifiée, registered in France · founded September 2022 · Paris & Rabat

Open-source reference stack, exploitation-surface paper, the registry pattern · all at **teams.cohorte.co/research**